*Training in **Cooperative Cyber Defence Centre of Excellence**, Tallinn, Estonia*

Scanning

Version 1.0, 2010-12-03, part of:
Introductory course in IT systems attack and defence

Kaur Kasak kaur.kasak@ccdcoe.org

Copyright statement

- This material is a product of the CCD COE.
- Reproduction of material is authorized, provided the source is acknowledged, unless it is stated otherwise. Where prior permission must be obtained for the reproduction or use of material. Enquiries regarding authorization for reproduction can be sent to CCDCOE address ccdcoe@ccdcoe.org.

Purpose of Scanning

- Scanning is used by system administrators, network engineers, auditors and security specialists for the following activities:
  - security testing and auditing
  - network asset management
  - compliance checking
- At the same time, network scanning is also used by the intruders to find interesting targets

  From attackers or pen-testers perspective the main goal of the scanning phase is to map out the target network topology and get a list of **potential vulnerabilities**

Phases of Scanning

1. **Host Discovery** – discovering live systems (workstations, servers, routers, switches, access points, etc)
2. **Port Scanning** – identifying open TCP and UDP ports
3. **Service and Application Version Detection** – identifying services and the versions of services that are listening on open ports
4. **Enumeration** – DNS names, user accounts and open shares
5. **OS Fingerprinting** – determining the operating system of alive computers
6. **Vulnerability Scanning** – identify known vulnerabilities in known network services and applications

DNS Enumeration

- DNS enumeration is the process of identifying all DNS servers for given domain and corresponding DNS **R**esource **R**ecords
- DNS enumeration is in reality more like an active reconnaissance technique. It is valuable to identify used IP addresses, DNS names could indicate the purpose of specific system (*fw, vpn*, *file*, *ids*, *log*, *intranet*, *win*, *linux*) and give out other information
- DNS Zone Transfer (AXFR, IXFR) – DNS database replication, e.g. from master to slave
  - The permission to conduct a zone transfer should be restricted only to specific hosts

DNS Enumeration Tools

- nslookup, dig: for custom DNS queries
  - dig @server name type
    - server: IP or name of the DNS server to query
    - name: name of the resource record
    - type: type of the query like A, MX, NS, AXFR, SOA
  - dig @8.8.8.8 www.ccdcoe.org NS
  - dig @ns.ccdcoe.org www.ccdcoe.org MX
- /pentest/enumeration/**fierce**
- /pentest/enumeration/**dnsenum**
- /pentest/enumeration/**dnsrecon**

DNS Enumeration Tools II

- Common steps for DNS reconnaissance tools (based on *fierce2*)
  - Find authoritative DNS servers for the target
    - NS, ARIN lookup
  - Try zone transfer on those servers
  - Brute-force names that are often used
  - Make reverse DNS lookups for found IP addresses
  - Brute-force TLDs: sometimes the target has registered many TLD like target.com, target.ch, target.eu
  - http://trac.assembla.com/fierce/wiki/Techniques
- ./dnsrecon.rb -b example.com hosts.txt

Network Scanning Tools

- Network Scanning:
  - **Nmap** Security Scanner
    - De facto standard network scanner
    - Examples of different scan types in the next slides will be mainly based on Nmap
  - Zenmap – official GUI for Nmap
  - Hping, unicornscan
- Vulnerability Scanning
  - OpenVAS - Open Vulnerability Assessment System
  - Metasploit scanning modules

Host Discovery

- There are several methods that could be utilized to determine alive hosts
  - ARP ping
  - ICMP ping
  - TCP ping
  - UDP ping
  - IP Protocol ping
- Nmap command to conduct only host discovery

root@bt:~# nmap -sn TARGET -v --packet-trace

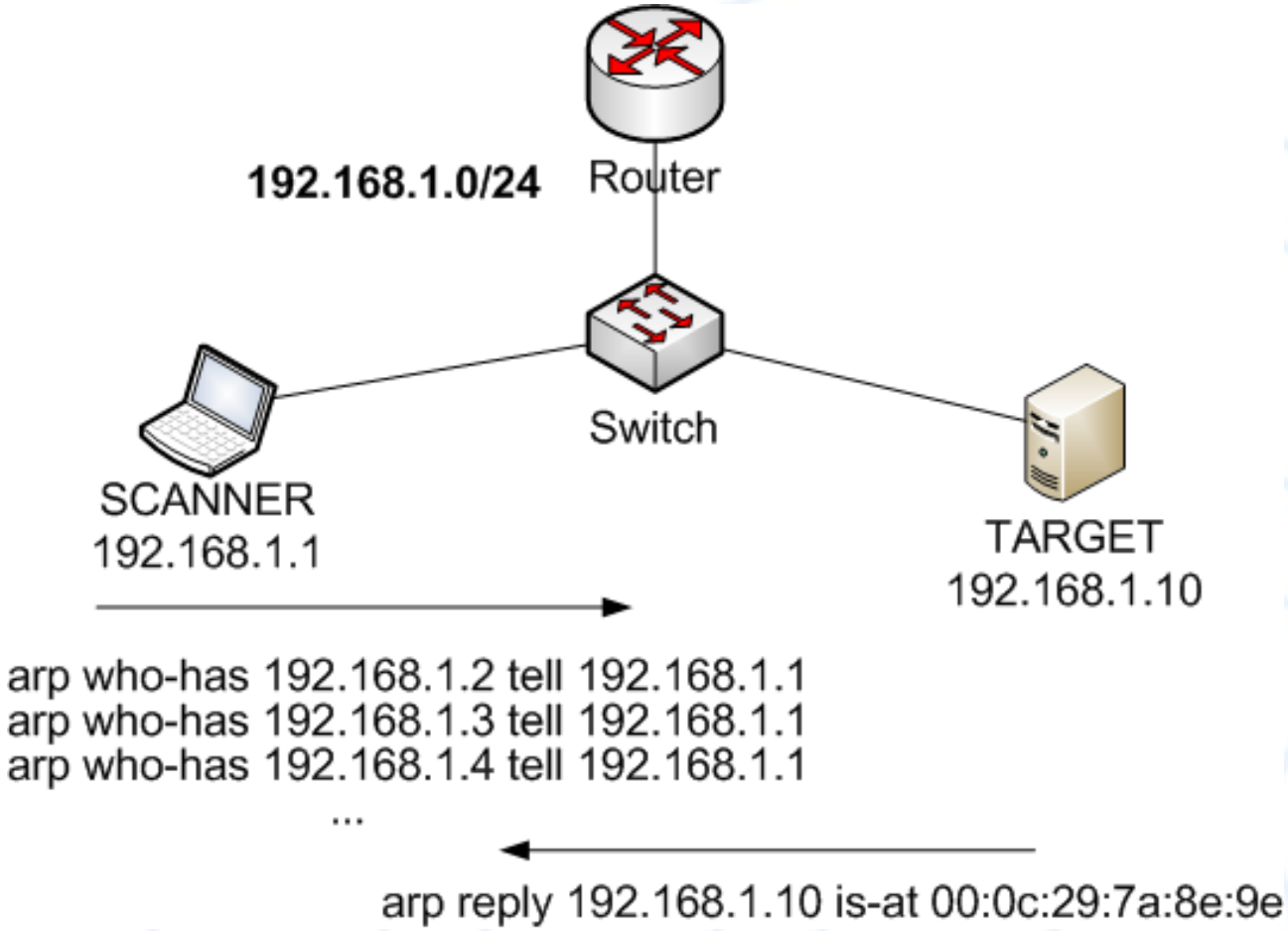
-sn:ICMP ECHO request, SYN TCP:443, ACK TCP:80, ICMP timestamp request

Host Discovery: **ARP ping**

- Address Resolution Protocol (ARP) is used to determine host's hardware address (Layer 2 address, MAC) when only the IP address is known
- When the computer is scanning a LAN to which it is directly connected an ARP scan gives usually the most reliable results about live hosts. Even heavily firewalled devices will typically respond to ARP probes
- By default Nmap uses only ARP scan when it is executed on the host that is located within the same network with the targets

```
root@bt:~# nmap -PR TARGET
```

Host Discovery: **ARP ping**



Host Discovery: **ICMP ping**

- Standard method to test connectivity with specific host is to send ICMP type 8 (Echo request) packets to the target
- If the target replies with ICMP type 0 (Echo reply) it is considered to be up
- This scanning method is not reliable for scanning Internet hosts because a lot of hosts and firewalls block these ICMP messages

```
root@bt:~# nmap -PE TARGET -v
```

- Other types of ICMP messages could be also used for host discovery (timestamp request, address mask request)

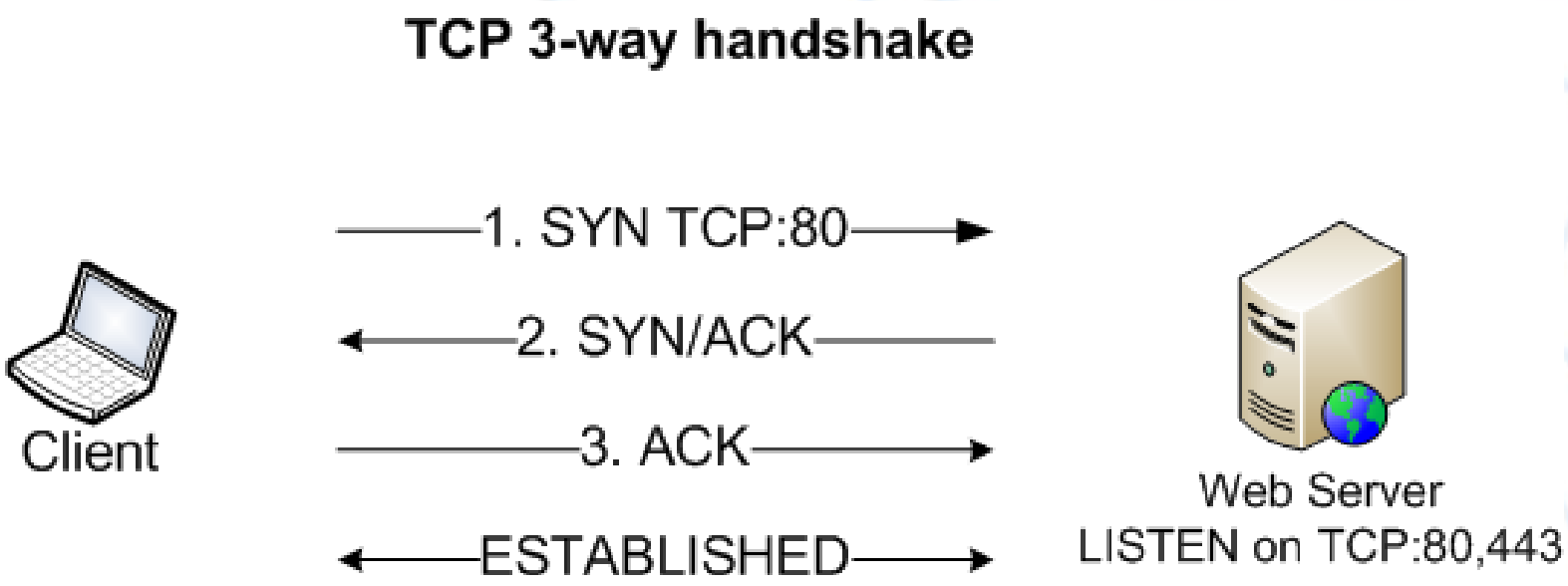
```
root@bt:~# nmap -PP -PM TARGET -v
```

Host Discovery: **TCP SYN ping**

- If ICMP protocol has been blocked by the network administrators other scan types have to be used
- **TCP SYN ping**
  - Send an empty TCP packet with SYN flag set
  - To speed up the process only the ports that are most probably open could be used e.g. 80, 23, 443, 21
  - If the target replies with
    - RST or SYN/ACK -> host is up
    - Nothing -> we do not know

```
root@bt:~# nmap -n -sP -PS80,443,3389 TARGET
```

TCP 3-way handshake



Host Discovery: **TCP ACK ping**

- Similar to TCP SYN ping
- Send a TCP packet with ACK flag set
  - This would acknowledge that bytes over an established TCP connection have been received
  - In reality, no connection has been established and the receiver should answer with RST if it is alive
- Reason for using TCP ACK in addition to TCP SYN is to maximize the chances of bypassing firewalls. Sometimes non-stateful firewalls have been configured only to drop incoming SYN packets but the packets with ACK flag set are let though

```
root@bt:~# nmap -n -sP -PA80,443,3389 TARGET
```

Host Discovery: **UDP ping**

- Send an empty UDP packet to the given port
  - If the port is closed but the target is up, it should send ICMP port unreachable (ICMP type 3 code 3) packet as a reply
- Primary purpose is to bypass firewalls

```
root@bt:~# nmap -n -sP -PU31313 TARGET
```

Port Scanning

- The purpose of port scanning is to identify the state of target's TCP and UDP ports
- Nmap classifies ports as **open**, closed, filtered, unfiltered, open|filtered, closed|filtered
- Port Scanning Techniques
  - TCP SYN scan
  - TCP Connect scan
  - TCP ACK
  - TCP FIN, Xmas, Null
  - UDP

TCP SYN Scan

- Most popular scan type: send TCP packet with SYN flag set to the target ports
- Fast, stealthier, reliable differentiation between the states
- **Privileges to create raw-packets required**

root@bt:~# nmap **-sS** TARGET -p1-1000 -r --packet-trace -v

Probe Response	Assigned State
TCP SYN/ACK	open
TCP RST	closed
No response	filtered
ICMP unreachable	filtered

TCP Connect Scan

- Mostly used when the user cannot create raw packets and connect() syscall has to be executed
- Performs a full 3-way handshake, then sends RST to kill connection
- Less efficient than SYN Scan
  - Takes longer
  - Target machines are more likely to log the connection (not so much difference in reality – IPS will catch both type of scans)
  - Generates more packets

root@bt:~# nmap **–sT** –n TARGET -p80,81

UDP Scan

- UDP scanning is a bit special - no connection is set up at the transport layer before application layer communication can occur
- Therefore UDP port scanning is sometimes ignored because it is slower and more difficult
- Many wide-spread and exploitable services still run on UDP e.g. DNS, NTP, SNMP, DHCP,...
- 2 main methods:
  - Send an empty UDP packet
  - Sending service specific probes (version scanning)

UDP Scan

- Scanning with an empty UDP packet

  root@bt:~# nmap -sU TARGET –p0-1024

Probe Response	Assigned State
Any UDP response	open
No response	open|filtered
ICMP port unreachable (type 3, code 3)	closed
Other ICMP unreachable	filtered

- Scanning with version specific probes

  root@bt:~# nmap -sUV TARGET -p0-1024 -r

Version Detection

- Detecting the version of the services and applications listening on open ports:
  - Service version could be useful to determine if it has any vulnerabilities
    - Version number does not indicate always if the vulnerability exists. Vendors could back-port security fixes without changing the version number
  - Service could be listening on non-default port
  - Un-registered ports
  - UDP port scan with version specific probes gives more accurate results

Version Detection with Nmap

- General Technique:
  1. Send Probe String
  2. Compare response to *match* strings and *softmatch* strings
  3. In case match, the scanning is completed. In case *softmatch* the scanning continues but is limited to probes that are known to match the given service.
- Firstly, a NULL probe is done: connect to the TCP port and wait for 6 seconds – many services identify themselves with a welcome banner

  /usr/share/nmap/nmap-service-probes

Version Detection with Nmap

- TCP services:

  root@bt:~# nmap -sV -Pn TARGET -p 70-90

- UDP services:

  root@bt:~# nmap -sUV -Pn TARGET -p 70-90

- To see the connection and read/write activities:

  root@bt:~# nmap -sV -Pn TARGET -p 70-90 **--version-trace**

- It is possible to control the intensity level of version detection (how many probes will be used):

  root@bt:~# nmap –sV TARGET **--version-intensity** 9

OS Fingerprinting

- Reasons for operating system detection:
  - Differentiate between devices: printer, router, switch, server, desktop:
    - Detecting unauthorized and dangerous devices
  - Identify effective exploits and real vulnerabilities, reducing false positives without necessarily testing the exploits out
  - Develop OS specific exploits – shellcode and payload has to take into account target's OS and hardware

OS Fingerprinting Methods

- TCP/IP stacks of different OS and different versions of the same OS act differently
  - TCP, UDP, ICMP probes could be sent to the target and fingerprint generated based on the responses
  - Nmap fingerprint database: /usr/share/nmap/**nmap-os-db**
- Passive fingerprinting
  - Sniffing network traffic and classifying hosts
  - **p0f** – passive OS fingerprinting tool
- Open port patterns
  - TCP 139, 445, 3389
- IP fragmentation

OS Fingerprinting

- Adding –O to Nmap scan options turns on OS detection

  root@bt:~# nmap **-O** TARGET –v –osscan-guess

  -v adds some extra lines to the results (device type, OS details, network distance)

  -osscan-guess in case there is no direct match, you may ask to print guesses

- If you have experience, then combinding OS detection with service version scan may also help:

  root@bt:~# nmap -sV **-O** TARGET –v

- You can also try Xprobe for OS detection:

  root@bt:~# xprobe2 TARGET -T 1-4000 -U 100-200 -v

Vulnerability Scanning

- Effective and fast way to determine security status of the systems
- Vulnerability scanners apply many of the methods already described: DNS enumeration, host discovery, port scanning, OS and service version detection
- Many vulnerabilities are still reported based on service version numbers. This method tends to create a lot of **false positives**
  - Not good idea to continuously provide reports full of false positives to your clients
  - Manual verification needed

Vulnerability Scanning

- Other kind of checks
  - Conducting registry and file checks by accessing the host if credentials provided and identifying missing patches or configuration mistakes
  - Trying actually to exploit the vulnerabilities to be sure the problem exists
    - Safe checks
    - Destructive tests
    - Problem of **false negatives** in case the attack script is flawed
  - Testing for web application vulnerabilities

Example: ms_telnet_overflow.nasl

```
##ATTACK####Vulnerability tested on AYT commands##
function telnet_attack(port){
  iac_ayt = raw_string(0xff, 0xf6);
  bomb_size = 100000;
  sock = open_sock_tcp(port);
  if(sock){
    bomb = crap(data:iac_ayt, length:2*bomb_size);
    send(socket:sock, data:bomb);
    close(sock);
    return(1);
  }else{
    return(0);
  }
}
```

Example: ms_telnet_overflow.nasl

```
##MAIN##
port = get_kb_item("Services/telnet");
if(!port) port = 23;

if(telnet_attack(port:port)){
  sock = open_sock_tcp(port);
  if(!sock){
    security_hole(port);
  }else{
    close(sock);
  }
}else exit(-1);
```

Vulnerability Scanning

- Keep in mind that the vulnerability scanners are just tools and the results need manual verification
- Network Vulnerability Scanners
  - Nessus, OpenVAS, Core Impact, Metasploit, Saint, Microsoft Baseline Security Analyzer
  - Nmap Scripting Engine
- Web Application Vulnerability Scanners
  - sqlmap, WebScarab, Nikto, AppScan...

Nmap Scripting Engine

- Allows to write simple scripts for automating different networking tasks
  - Information Gathering from 3rd party databases
  - Network Discovery: enumerating SMB shares, looking up SMB data, SNMP queries
  - Vulnerability detection and exploitation
  - Malware detection on target host

http://nmap.org/nsedoc/

- Example: scanning for open proxies:

nmap -n **-sV -sC --script http-open-proxy** -p 8080 **TARGET** -v

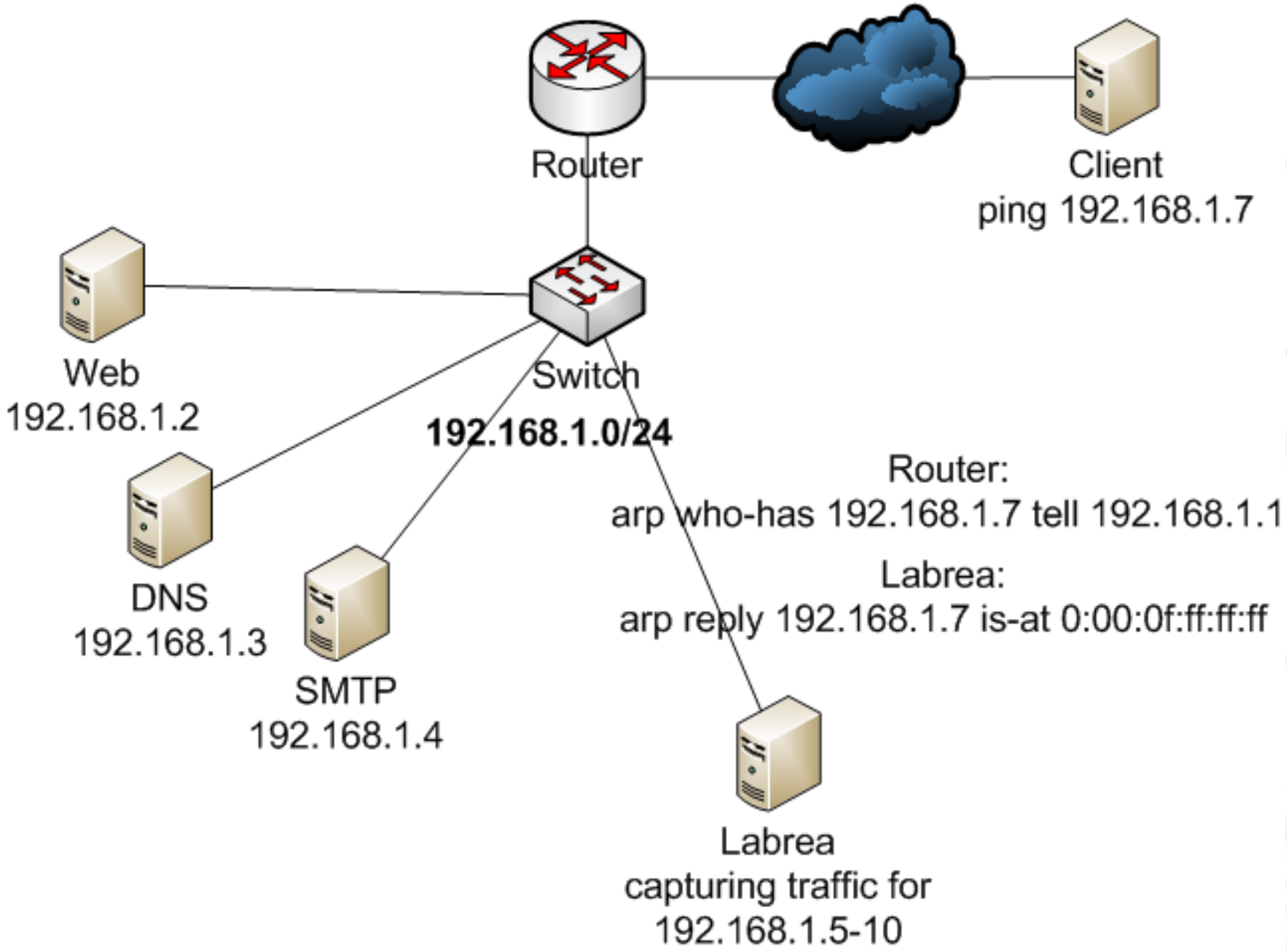
Defenses Against Scanning

- Scan your own networks, disable services not required, patch,...
- Use well-configured firewalls
- Detect scans
- Deceiving and confusing attackers. Obscurity
  - Honeypots and Honeynets
  - Tarpits
  - “Hiding” services on uncommon ports
  - Port Knocking
  - Modifying banners
  - OS hardening and OS version obfuscation

Labrea Tarpit

- "Sticky honeypot"
- Creates virtual systems on unused IP addresses
  - Labrea answers to the ARP **who-has** requests with it's own IP
  - Replies to incoming SYN packets with SYN/ACK. Does not reply to other attempts to send additional data

    Sender (potential scanner) tries to retransmit
  - Replies to ICMP ECHO REQUEST with ICMP ECHO REPLY
  - Other activities



Labrea Tarpit

- TCP protocol defines **window size** for Flow Control
  - Receiver specifies receive window size: how much bytes it is able to buffer at once
  - Sender can send only this amount of data before it must wait ACK from the receiver
- Labrea exploits this behavior to slow the scanners:

TCP 55047 > 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460

TCP 80 > 55047 [SYN, ACK] Seq=0 Ack=1 Win=5 Len=0

TCP 55047 > 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0

HTTP [TCP Retransmission] Continuation or non-HTTP traffic

HTTP [TCP Retransmission] Continuation or non-HTTP traffic

Nmap useful options

- Specifying target hosts and networks
  - CIDR style: 192.168.0.0/16
  - Comma-separated list: 192.168.0-127.1-10
  - Input from file:-iL <input filename>
- Misc:

-n: no DNS resolution

-v: increase verbosity level

--reason: Display the reason a port is in a particular state

--packet-trace: Show all packets sent and received

--script-trace: Show all data sent and received during script scan

Nmap useful options II

- nmap –A –T5 TARGET

  -A: enables number of advanced options like OS detection, version detection and NSE

  -T5: use aggressive timing template

OpenVAS

To start OpenVAS vulnerability scanner first time on **BackTrack**, execute the following commands:

- openvas-mkcrt
- openvas-adduser
- openvassd
- OpenVas-Client &
- Task -> Scope -> Connect -> Choose plugins and target -> Execute

Note that on the students lab machine the plugins have been already updated. Do not try to update again because vm disk is small but OpenVAS will create thousands of small files consuming all the inodes!

Introductory course in IT systems attacks and defence

PRACTICAL EXERCISES IN NETWORK SCANNING

References

- Gordon Lyon. Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure.Com LLC, 2008.
- http://nmap.org/presentations/BHDC10/Nmap-NSE-Fyodor-David-Defcon18-HD.mov